DTIC FILE COPY

AD-A218 976

COGNITIVE EFFICIENCY CONSIDERATIONS FOR GOOD GRAPHIC DESIGN

Technical Report AIP - 81

Stephen Casner
University of Pittsburgh

Jill H. Larkin
Carnegie Mellon University

1989

The Artificial Intelligence and Psychology Project

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

DTIC ELECTE MAR 13 1990 S D B

Approved for public release; distribution unlimited.

90 03 12 038

COGNITIVE EFFICIENCY CONSIDERATIONS FOR GOOD GRAPHIC DESIGN

Technical Report AIP - 81

Stephen Casner
University of Pittsburgh

Jill H. Larkin
Carnegie Mellon University

1989

This research was supported by the Computer Sciences Division, Office of Naval Research, under contract number N00014-86-K-0678. Reproduction in whole or part is permitted for any purpose of the United States Government. Approved for public release; distribution unlimited.

DTIC
ELECTE
MAR 13 1990
S D
B

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP - 81	5. MONITORING ORGANIZATION REPORT NUMBER(S) Same as Performing Organization
6a. NAME OF PERFORMING ORGANIZATION Carnegie Mellon University 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Personnel and Training Research Office of Naval Research (Code 1142PT)
6c. ADDRESS (City, State, and ZIP Code) Department of Psychology Pittsburgh, Pennsylvania 15213	7b ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, VA 22217-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as Monitoring Organization 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-88-K-0086
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: N/A; PROJECT NO.: N/A; TASK NO.: N/A; WORK UNIT ACCESSION NO: N/A

11 TITLE (Include Security Classification)

Cognitive Efficiency Considerations for Good Graphic Design

12 PERSONAL AUTHOR(S)

Stephen Casner and Jill H. Larkin

13a. TYPE OF REPORT	13b. TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 86Sept 15 TO 91Sept 14	1989	7

16 SUPPLEMENTARY NOTATION

17 COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
GROUP	SUB-GROUP	graphic design, user interface, task analysis, visual languages

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

SEE REVERSE SIDE

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION
22 NAME OF RESPONSIBLE INDIVIDUAL Susan Chipman	22b TELEPHONE (Include Area Code) (202) 696-4322 22c. OFFICE SYMBOL 1142 PT

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

ABSTRACT

Larkin and Simon's (1987) analysis of how graphical representations support task performance is applied to designing graphical displays that streamline information-processing tasks. Theoretically this streamlining is done by designing external data structures that (a) allow users to substitute less effortful visual operators for more effortful logical operators, and (b) reduce search for needed information. A design program called BOZ is used to produce four alternative displays of airline schedule information to support an airline reservation task. We postulate several procedures that use a set of visual operators to perform the task using the different graphics. The number of times each operator is executed provides one measure of task difficulty (for a procedure and graphic). A second measure is the difficulty of executing each operator. Seven subjects performed the airline reservation task using each of the four graphics. Response times for the different graphics differ by a factor of two, and this defference is statisically highly significant. Detailed data analyses suggest that these differences arise through substitution of visual operators for logical ones and through the use of visual cues that help reduce search. The analyses provide quantitative estimates of the time saved through operator substitutions.

QUALITY INSPECTED 4

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

Cognitive Efficiency Considerations for Good Graphic Design

Stephen Casner
University of Pittsburgh

Jill H. Larkin
Carnegie Mellon University

Abstract

Larkin and Simon's (1987) analysis of how graphical representations support task performance is applied to designing graphical displays that streamline information-processing tasks. Theoretically this streamlining is done by designing external data structures that (a) allow users to substitute less effortful visual operators for more effortful logical operators, and (b) reduce search for needed information. A design program called BOZ is used to produce four alternative displays of airline schedule information to support an airline reservation task. We postulate several procedures that use a set of visual operators to perform the task using the different graphics. The number of times each operator is executed provides one measure of task difficulty (for a procedure and graphic). A second measure is the difficulty of executing each operator. Seven subjects performed the airline reservation task using each of the four graphics. Response times for the different graphics differ by a factor of two, and this difference is statistically highly significant. Detailed data analyses suggest that these differences arise through substitution of visual operators for logical ones and through the use of visual cues that help reduce search. The analyses provide quantitative estimates of the time saved through operator substitutions.

1. Introduction

Empirical studies of graphics find little support for any general superiority of graphical representations. Instead, graphic displays seem to vary in usefulness depending on the task involved. Twenty-nine studies (Jarvenpaa and Dickson,1988) found graphics to be more useful than tabular presentations for some tasks, but less useful for others. These results are consistent with the theoretical analysis of Larkin and Simon (1987) that a display (graphic or otherwise) is a data structure. It's utility depends on the nature of the task it supports and the nature of the procedures used to perform the task. When procedures and data structures match well, there is better cognitive efficiency than when they do not.

Larkin and Simon (1987) suggest that the following forms of cognitive efficiency are offered by good graphical displays.

- **Substituting Visual Operators:** Graphical displays often allow users to substitute less demanding visual operators in place of more complex logical operators. Visual operators (e.g., distance and color comparisons, spatial coincidence judgements) can often give users the same information as more complex non-visual operators.
- **Reducing Search:** Graphical displays often arrange information in such a way as to reduce the number of items the user must look at in order to find something useful, or to group information required to draw a particular inference into one spatial locality. Graphical techniques like shading and spatial arrangement can help guide the eye to relevant information or past irrelevant information.

This paper describes BOZ, a computer-implemented algorithm for designing graphical displays (Casner, 1989). BOZ (described in Section 2) systematically exploits the hypothesized advantages of graphical displays, substituting visual operators for logical ones, and constraining the grouping of related information. BOZ analyzes a formal description of the operators that are required to execute a task and searches a catalog of visual operators to find visual operators that can serve as substitutes for the logical operators. BOZ then proposes graphic displays that support performance of these operators. A single task description typically gives rise to many graphic displays, each supporting different substitutions of visual for logical operators. Section 3 describes four alternative graphical displays proposed by BOZ to support the task of finding airline reservations satisfying time and cost constraints. We hypothesize

several visual procedures that can be used to perform the reservation task with the four graphics. A simulation counts the number of times each visual operator executes for each combination of procedure and graphic. Section 4 describes an experiment in which participants used the four BOZ-designed graphics. Comparisons of participants' response times against the operator counts suggest two mechanisms through which these graphics improve cognitive efficiency. (1) substituting visual operators for logical ones, and (2) reducing search by using visual cues that allow certain items to be ignored.

2. BOZ: Designing Effective Visual Data Structures and Procedures

A Logical Operator Description Language. BOZ (Casner, 1989) begins with a description of the set of logical operators (LOPs) required to perform task. Logical operators are general information-processing activities independent of any particular representation. To use BOZ to design a graphical display, an encoded description of the relevant logical operators is submitted to the algorithm. For example, the following LOP describes finding the layover between two connecting airline flights:

```
(LOP findLayover (flightA flightB)
    (DIFFERENCE
        (findDeparture flightB)
        (findArrival flightA)))
```

findLayover takes two flights as arguments and computes returns the layover time—the difference between the arrival time of flightA and the departure time flightB.

A Catalog of Visual Operators. BOZ contains a catalog of visual operators that describe information-processing activities that occur within the context of a graphical display. Visual (or perceptual) operators (POPs) include spatial position and coincidence judgements, interval and distance judgements, comparisons of color, shape, size, slope, length, height, width, etc. POPs are encoded using the same formalism used to describe LOPs. For example, the operator for estimating horizontal distance between two graphical objects is:

```
(POP findHorzDistance (objA objB)
    (DIFFERENCE
        (findHorzPos objA)
        (findHorzPos objB)))
```

Matching Logical Operators to Visual Operators in the Catalog. A matching algorithm considers each logical operator in a task description and searches the catalog of visual operators for substitutes. A visual operator qualifies as a substitute if renaming can map the visual operator into the logical operator. For example, findHorzDistance and findLayover are equivalent because they both compute a difference between two numbers (flight times and horizontal coordinates). Although not discussed here, often no single visual operator matches a LOP. BOZ then attempts to match more complex logical operators using two or more visual operators and a set of combination, composition, and repetition rules.

Visually Structuring Related Data. For each proposed substitution of visual for logical operator, a data structuring algorithm assesses the information required to perform the operator and tries to ensure that this information is presented in the same spatial locality and in a form that supports easy perceptual performance of that visual operator. For example, if findHorzDistance replaces findLayover, then the data structuring algorithm requires that. (1) all times are encoded along the same axis, allowing a human to substitute estimating horizontal distance between two objects for the logical operator of subtracting their coordinates; and (2) all time information about a single flight is encoded using the same graphical object.

3. An Example: Graphical Displays for Airline Reservations

We used BOZ to design a set of graphical displays to support the following airline reservation task that manipulates information about flights, their origins and destinations, departure and arrival times, and costs

Find a pair of connecting flights that travel from Pittsburgh to Mexico City. You are free to choose any intermediate city as long as the layover in that city is no more than four hours. Both flights that you choose must be available. The combined cost of the flights cannot exceed $500.

The task description submitted to BOZ contained the following logical operators:

findFlight(endpoint, city)
Sequentially searches a list for a flight with **endpoint** (origin or destination) equal to **city**. Returns the first flight meeting this criterion.

checkAvailability(flight)
Returns "true" if a flight has seats available.

checkLayover(flightA, flightB)
Returns "true" if the layover between two flights is acceptable (non-negative and less than 4 hours).

checkCost(flightA, flightB)
Returns "true" if the cost of the two flights is acceptable (less than $500).

Figure 1 shows four displays produced by BOZ when given the description of these logical operators. We consider these displays in turn, describing how BOZ created them, and correspondingly their hypothesized advantages to a user.

Display 1: A Conventional Airline Schedule
Display 1 is a tabular presentation that supports substituting a set of visual operators pertaining to tables for the corresponding logical operators listed above.

findFlight(endpoint, city)
search the rows of the table stopping at a row that has an endpoint (origin or destination) equal to **city**.

readAvailability(flight)
true if second column reads "ok"; else false

subtractTimes(flight1, flight2)
find departure time in the flight2 row (column 4) and arrival time in the flight1 row (column 5);
subtract arrival time from departure time;
return true if greater than zero and less than 4 hours; else false

addCosts(flight1, flight2)
find cost in flight2 row (column 3) and cost in flight1 row (column 3);
add the two costs;
return true if less than $500; else false

We define the following search procedure, called **rowSearch**, that uses the four operators and sequentially considers flights in the order that they appear in the rows of the table. Aside from exploiting the row and column indexing of information, the rowSearch procedure is essentially non-visual.

```
procedure rowSearch
    repeat
        findFlight(flight1, origin, pit)
        if   readAvailability(flight1) then:
             findFlight(flight2, CITY, mex)
        if   readAvailability(flight2) then:
        if   subtractTimes(flight1, flight2) then:
        if   addCosts(flight1, flight2) then:
             report answer
    until answer found
```

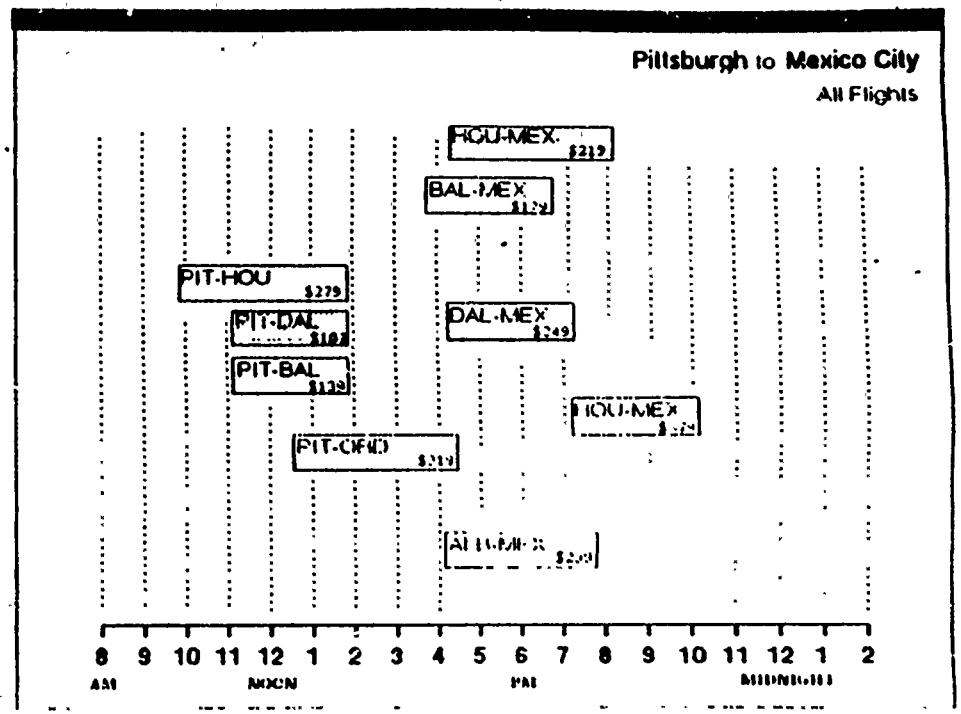
DISPLAY 1

Pittsburgh to Mexico City

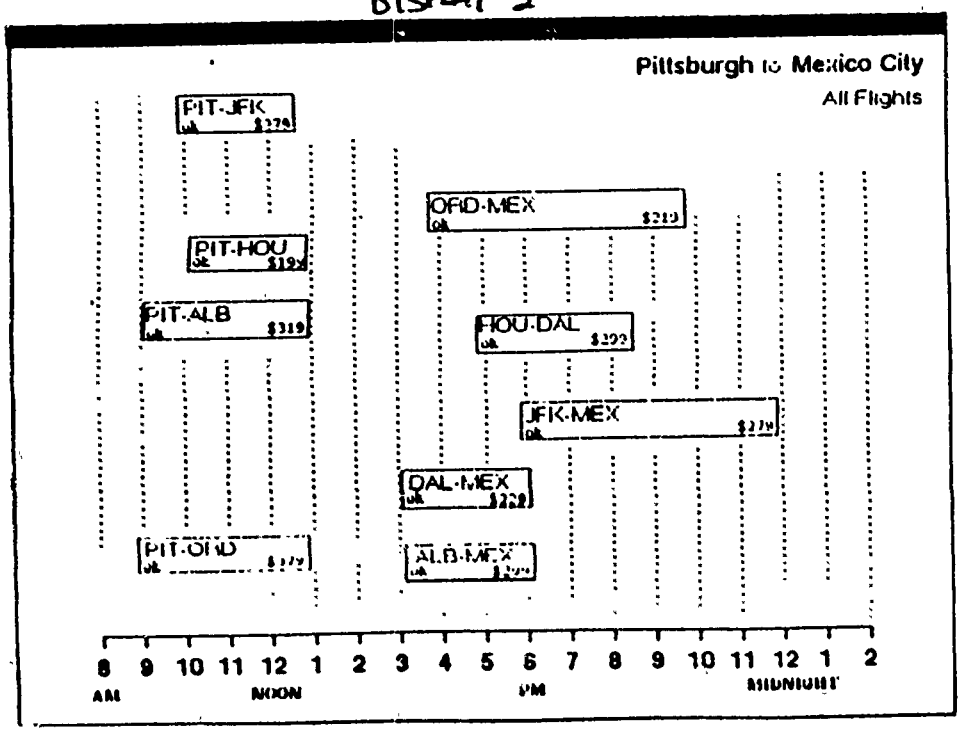
All Flights

Origin/ Destination	Availability	Price	Departure	Arrival
PIT-ORD	ok	$179	10:00am	12:00pm
PIT-HOU	ok	$99	9:15am	12:05pm
ALB-MEX	full	$279	3:50pm	9:50pm
PIT-JFK	ok	$439	9:50am	12:30pm
ORD-MEX	ok	$319	4:15pm	10:50pm
ORD-DAL	full	$199	1:00pm	4:30pm
DAL-MEX	ok	$229	7:30pm	10:30pm
PIT-ALB	ok	$219	8:00am	11:50am
HOU-MEX	ok	$229	3:00pm	6:00pm

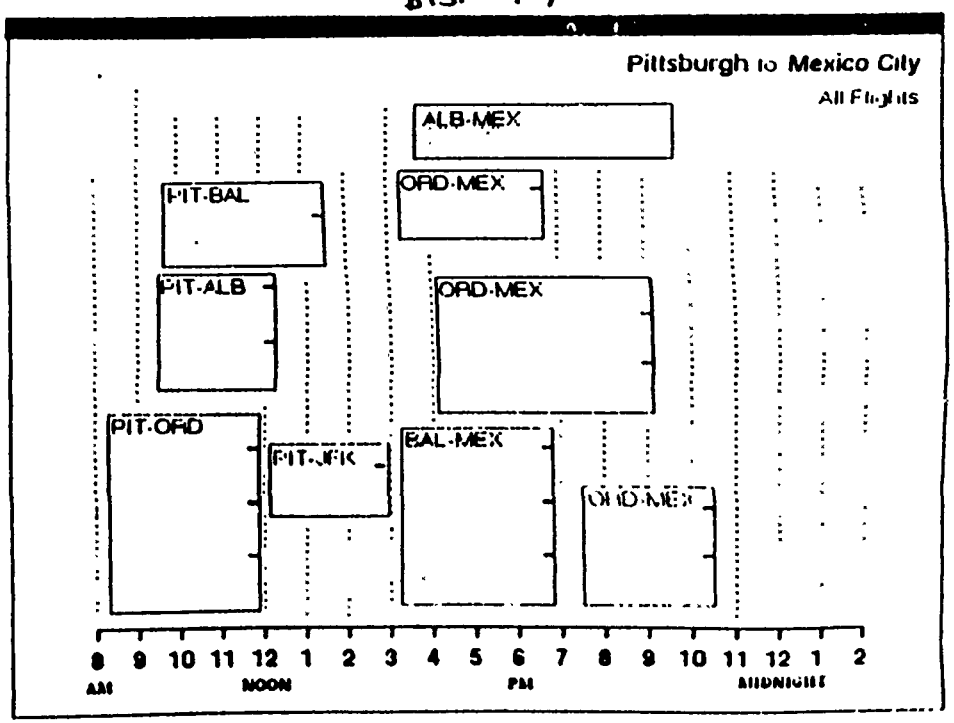
DISPLAY 3



DISPLAY 2



DISPLAY 4



Display 2: Horizontal Distance Encodes Times
Display 2 substitutes the visual findHorzDistance operator for the essentially logical subtractTimes operator. If two connecting flights have ends within four units of each other, then the layover is less than four hours. As required by the data structuring algorithm, all times are encoded as horizontal positions, and the two times associated with one flight are encoded by the same graphical object, i.e., a box.

Display 2 also supports two variations of the rowSearch procedure. In **rightOfSearch**, the findFlight operator considers only those flight boxes appearing to the right of an originating flight box. rightOfSearch is the same as rowSearch, but omits consideration of flight boxes that are not to the right of the end of the current flight box. In **closeSearch**, users first consider those pairs of flights that are closest together (have the shortest layovers).

Display 3: Shading Encodes Availability
Display 3 substitutes the perceptual task of determining the shade of a flight box for the readAvailability operator. Additionally, Display 3 lets users modify any search procedure by skipping all shaded flight boxes. These procedures are indicated by **rowSearchU**, **rightOfSearchU**, and **closeSearchU**, where the final "U" indicates searching only unshaded boxes.

Display 4: Height Encodes Cost
Display 4 allows users to substitute the perceptual task of judging the combined heights of two flight boxes for the addCosts operator. Display 4 supports the **cheapSearch** and **cheapSearchU** procedures which consider the cheapest (least tall) flight boxes first.

Display Summary
Sequenced as in Figures 1, each display, compared to the preceding one, supports (a) substituting one additional visual operator, and (b) reducing search by pruning visually detected violations of that operator. In summary, these cumulative advantages are:

Graphic	Visual operator substitution	Improved search procedures
1	<<direct analogs of logical operators>>	rowSearch
2	subtractTimes—>findHorzDistance	rightOfSearch, closeSearch
3	readAvailability—>checkShading	rowSearchU, rightOfSearchU, closeSearchU
4	addCosts—>stackHeights	cheapSearch, cheapSearchU

4. Empirical test of design effectiveness

Method

Participants. Seven employees of the Learning Research and Development Center at the University of Pittsburgh.

Materials. There were a total of 40 problems, ten different instances of each of the four displays. Examples of each of the four displays are shown in Figure 1. The airline information appearing in the problems was equalized such that the total number of times the operators were executed was the same for each of the four display versions with respect to any procedure followed.

Apparatus. Displays were presented as 9 x 12 inch screen images on a Xerox 1186 computer. Response times were computed using the system clock when the mouse was clicked.

Procedure. Participants performed the task forty times, ten times using each display versions. The order of the versions was varied systematically to counterbalance learning and practice. Eight orders were used (one for each participant), four rotations of forward order (1,2,3,4), (2,3,4,1) ...), and four of backwards order (e.g., (4 3 2 1), (3 2 1 4), ...). One participant's data is missing from the analysis. At the start of the experiment, all relevant visual operators were explained. Participants were shown the rowSearch

procedure but were told that they could follow any strategy they wished. There was one practice trial with each version. Participants were told not to guess, to work as quickly as possible but not to compromise accuracy, and that they could rest between any two graphics. Time to complete the experiment was typically 40 minutes.

Empirical Predictions

Global Efficiency. Each graphic supports the advantages of the previous one, as well as the ones it introduces. Therefore the first prediction is that cognitive efficiency should be linearly ordered as in Figure 1 with the conventional table worst and Display 4 best.

Decrease in operator times. For every combination of display and search procedure, we can count the number of times each operator is executed. If response times are expressed as a function of the number of executions of each operator, a regression analysis yields estimates of the times associated with each operator. If substituting visual operators for logical ones improves efficiency, then the times associated with the "computation" operators (checkAvailable, checkLayover, and checkCost) should be smaller for graphics that support substitution of visual operators.

Theoretical relations between search procedures. To compare search procedures we constructed ten instances of each display type. We built a LISP simulation to count the number of operator executions when any search procedure is followed on a particular display. The results of this simulation verify that there are three sets of very similar search procedures, those based on rowSearch, those based on closeSearch, and those based on cheapSearch.

RowSearch is the basis for rightOfSearch and rightOfSearchU. The relation between the numbers N_{row} and N_{rt} of search steps in rowSearch and rightOfSearch is reflected by the regression equation:

$$N_{rt} = .66\, N_{row} + 1.9,$$

for N_{row} between 10 and 60. $R^2 = .874$. rightOfSearch thus very consistently requires about two-thirds the amount of search as rowSearch. There are similarly close relations between the number of search steps required by the related procedures rowSearchU and rightOfSearchU.

$$N_{row,u} = .737N_{row} - .806,\ R^2 = .703$$

$$N_{rt,u} = .646N_{rt} + 2.799,\ R^2 = .502$$

Skipping unshaded boxes reduces search, but does not change the structure of any algorithm.

For the same 40 examples, the correlation between N_{row} and the number of search steps executions by closeSearch has $R^2 = .058$, indicating essentially no relation between the two quantities.

CloseSearch and closeSearchU are directly related,

$$N_{cl,u} = .741N_{cl} + 1.028,\ R^2 = .446 \quad \text{(based on first 30 displays)}$$

Similarly, the cheapSearch and cheapSearchU procedures are related to each other. But the closeFirst procedure are not related to the rowSearch procedures ($R^2 = .001$ for closeFirst and rowSearch), and cheapSearch is similarly not related to either rowSearch or closeSearch.

Preliminary comparisons of the three procedure groups (rowSearch, closeSearch, and cheapSearch) with the subject response times for each of the four displays suggest that only the rowSearch procedures provide reasonable fits to the data. Thus it seems that, with the practice available, subjects did not adopt the more sophisticated search strategies.

Results and Discussion

Global Efficiency. The top row of Table 1 shows the mean response times for each graphic (excluding five times differing by more than three standard deviations from the problem mean and the 3 to 6 erroneous responses for each graphic).

Table 1: Response Time Means and Standard Errors for the four graphics versions.

Graphic Version:	1. Table	2. Horizontal distance encodes times	3. Shading encodes availability	4. Height encodes cost
Mean Response Time (sec)	19.3	10.1	7.2	7.4
Standard Error of the Mean	8.4	4.7	2.7	2.4

Graphic version had a highly significant effect on response time ($F(3, 239) = 52.719$, $p < .0001$), and also on the variance of response time ($F(3, 24) = 18.649$, $p < .0001$). Table 2 indicates significant pairwise comparisons. Graphics 3 and 4 produce both the lowest response times and the least variable performance. Graphics 2 and 1 each in turn produce significantly higher response times and greater variability.

Table 2: Fischer's PLSD for pairwise comparison of
(a) mean response times and (b) standard errors of the mean for response times.
*'s indicate significant differences.

	(a) Graphic Version			(b) Graphic Version		
	2	3	4	2	3	4
1	*	*	*	*	*	*
2		*	*		*	*
3						

In summary, the visual operators supported by Displays 2 and 3 had the predicted effect on global efficiency. Allowing users to perform stackHeights produced no observable effect. This should perhaps not surprise us since it is the one visual operator that requires integrating quantitative estimates from two different locations.

Decrease In Operator Times. To assess the effect of substitution of visual for logical operators, we analysed the reaction times in the following way. We assumed that for each graphic the search procedure was the most efficient rowSearch procedure supported by that graphic, i.e., rowSearch for the table rightSearch for display 2 (with flight boxes), rightSearchu for displays 3 and 4 (with shaded boxes). We also postulated two alternative operators for assessing layover and cost. The subtractTimes and addCosts operators correspond to subtracting or adding numbers. These operators were assumed for displays that did not support alternative procedures (the table for subtractTimes, and displays 1, 2, and 3 for addCosts). For the remaining displays, we assumed use of the more efficient visual operators findHorizontalDistance and stackBox. We assumed that the time for one search step was the same in all graphics (although the number of such steps varied with the search procedure supported).

Using these assumptions we computed for each of the 40 graphic exemplars the number of search steps and the number of cost and layover computations. A regression of response times on these numbers produced a well-fitting statistical model with $F(4, 238)=73.108$, $p = .0001$, $R^2 = .48$. Removing from the model the counts for either search or checking layovers dramatically reduced the fit. In contrast, removing the counts for checking costs had no effect on the fit. This model yielded the following parameter estimates:

- One search step requires 330 ± 35 milliseconds.
- The findHorizontalDistance operator is 2 ± .25 seconds faster than the subtractTime operator.
- The stackBox operator is negligibly (100 ± 300 milliseconds) slower than the addCost operator.

These results are consistent with the global time differences in Table 1. The reduced performance time with each successive graphic arises for two reasons. First, attending only to boxes to the right of the current box and to unshaded boxes reduces the number of items that must be searched. Second, substitution of findHorizontalDistance for subtractTimes produces a substantial saving in time. In contrast stackBox, which requires integrating visual information from two separate locations, provides no such advantage. These two effects are sufficient to account for the response time differences in Table 1 (and for the lack of difference between Displays 3 and 4).

Summary

BOZ is a computer algorithm that starts with a task description and designs graphic displays supporting substitution of visual operators for logical ones, and pruning of search through visual cues. In an initial experimental test, four BOZ-designed graphics each included one additional visual operator and corresponding opportunities for pruning search, by using visual cues to ignore certain items or by restructuring search to consider more promising items earlier. Analysis of subjects' response times indicate strongly that two out of these three enhancements dramatically and significantly improved response times to the task. The unhelpful enhancement required integration of information from two separate locations. More detailed analyses suggest that these improvements were due to operator substitution and pruning of search through visual cues, but not due to restructuring search.

Acknowledgements

This work is supported Office of Naval Research, University Research Initiative Contract Number N00014-86-K-0678 and by a grant from the James S. McDonnell Foundation to the second author. William Oliver and Stellan Ohlsson provided helpful comments.

References

Casner, Stephen, "A cognitive approach to designing effective graphical illustrations," Learning Research and Development Technical Report, March, 1989.

Jarvenpaa, S.L. and G.W. Dickson, "Graphics and managerial decision making: Research Based Guidelines," *Communications of the ACM* **31**, 6 (June 1988), 764-774.

Larkin, Jill and Herbert Simon, "Why a diagram is (sometimes) worth 10,000 words," *Cognitive Science* 11, 1987, 65-99.